CHAPTER 5

TEXTURE EDGE DETECTION

5.1 INTRODUCTION

The effective usage of the texture primitive spectrum in performing texture segmentation is discussed in this chapter. Image segmentation is the partition of an image into different regions, each having certain properties. It is the first step of image analysis, which aims at either a description of the image, or a classification of the image. Fu and Mui (1981) categorized the image segmentation techniques into three classes: i) characteristic feature thresholding or clustering, ii) edge detection, and iii) region extraction. The abundance of approaches that can be found in the literature points out that there is no universal method (available for image segmentation) which will work for all images. However, almost all segmentation algorithms are either based on concepts of similarity (e.g. characteristics feature clustering) or discontinuity (e.g. edge detection).

In general, the edge or boundary is the place where there is a more or less abrupt change in gray level. Since most of the useful information of an image lies on the boundaries between different regions, the human visual system appears to make use of edge detection for the perception of a composite image. For conventional un-textured images several schemes have been suggested to detect the boundaries. Popular operators include Robert's, Kirsch's, Sobel's and Prewitt's and functional approximation methods. These conventional edge detectors assume homogeneous brightness in different

regions and that the boundary pixels have discontinuities in an intensity profile. Hence they are suitable for un-textured images. When applied to images with textured regions, conventional edge detectors fail to segment the textured image into heterogeneously textured regions. Instead they tend to detect the micro edges present within the textured regions. Hence, texture edge detection requires a mechanism, which can discriminate between the heterogeneous textural patterns in a composite image, not the discontinuity of gray levels. This necessitates extracting suitable texture descriptors over a substantial spatial support and defining a measure of discontinuity in texture features. The texture features must be superimposed with conventional edge detectors to detect the boundaries present between different textured regions. The usage of the proposed texture primitive spectrum for performing supervised and unsupervised classification has been highlighted in the previous chapter. The texture primitive spectrum as a global descriptor for representing texture image and its usage for performing the segmentation by edge detection is explained in this chapter. The edge detection process is illustrated in two different classes of texture images, namely, (i) Deterministic Textured Images and (ii) Non- Deterministic Textured Images.

5.2 VARIOUS APPROACHES

Unlike conventional edge detection techniques, which aim to detect the discontinuities in pixel intensity for locating the edges, texture edge detection methods look for discontinuities in local texture properties, defined over some neighborhood. Davis and Mitchie (1980) designed a procedure based on a class of one-dimensional contrast operators.

Rosenfeld et al (1976) has suggested a method in which the average gray level values are calculated in neighboring windows. If the difference between the values is higher, an edge point is located. The problem with this

method is that different texture primitives can have the same average gray level but with different spatial arrangement of pixels. In that case, an edge present will be left undetected. Moreover, the presence of noise pixels in the image will affect not only the statistical parameters, but also the detection of edge points (Rosenfeld et al 1976). Another method suggested is based on the statistical information derived from the textured regions. It is based on the Fisher Distance between the adjacent windows having local mean and standard deviation of gray levels μ_1,σ_1 and μ_2,σ_2 respectively. Fisher distance is calculated as follows,

$$D_F = \frac{|\mu_1 - \mu_2|}{\sigma_1 + \sigma_2} \tag{5.2}$$

If D_F is greater than a threshold, an edge point is marked. The restriction imposed in this method is that σ_1, σ_2 shall never be equal to zero. Sadler (1993) approached the texture edge detection problem via non-parametric change detection. Edges are those locations where the mean, variance or other second or higher order statistical features might undergo a change. The detection of such a change is then linked to edge detection. The features used for this purpose are 'texture energy' measures, which are derived by convolving the image by following Laws' [1980] micro-texture convolution masks,

$$L5S5 = \begin{bmatrix} -1 & 0 & 2 & 0 & -1 \\ -4 & 0 & 8 & 0 & -4 \\ -6 & 0 & 12 & 0 & -6 \\ -4 & 0 & 8 & 0 & -4 \\ -1 & 0 & 2 & 0 & -1 \end{bmatrix}$$

$$E5S5 = \begin{bmatrix} -1 & 0 & 2 & 0 & -1 \\ -2 & 0 & 4 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & -4 & 0 & 2 \\ 1 & 0 & -2 & 0 & 1 \end{bmatrix} \tag{3.3}$$

Laws' texture masks are applied on the image I(m, n) resulting in an output image c(m, n). The variance of c(m, n) is estimated to yield another image v(m, n). The variance image is smoothened to eliminate any undesirable peaks produced due to 'within the class' variations. Then, a one-dimensional change detection is applied on v(m, n) row-wise and column-wise to estimate the texture edges satisfying certain conditions. In another work, Ganesan and Bhattacharya (1997) presented a method capable of detecting edges both in textured and un-textured images. A complete set of difference of operators configured from orthogonal polynomials is employed to separate the responses toward edge or texture and noise.

Liu and Pok (1999) employed a set of Gabor filters by varying the center frequency and orientation to represent the texture features. A one-dimensional feature map over the filter energy output is generated by the Self Organizing Feature Map (SOFM) algorithm, which preserves the topologically ordered relations. The feature map 'F' is encoded as scalars by replacing the Gabor features by the corresponding location index on 'F'. Then a predictive relationship between an encoded feature and its neighbors is computed along eight directions by utilizing the function approximating property of the multilayer perceptron. The variance of eight prediction coefficients is used to represent the similarity of textural patterns around a given pixel. After Gaussian smoothing is applied over the variances to suppress local fluctuations, the edges are located by thresholding the gradient magnitude obtained using Canny's gradient operator.

Moment functions have been used as shape descriptors in a variety of applications in image analysis, like visual pattern recognition, object classification, template matching, edge detection, data compression etc. Works on using moments for texture characterization or representation have been written by (Bharathi et al 2002). The idea of using orthogonal moments for texture edge detection has been motivated by the fact that orthogonal moments have superior feature representation capability and low information redundancy. In this work, Zernike moments and Legendre moments have been employed to detect edges in heterogeneously textured images. The magnitudes of the Zernike moments or the Legendre moments are computed over an 8 × 8-pixel size window, centered at a pixel location of I(i, j). The number of moments used in the experiments is selected through a systematic procedure in order to reduce the dimension of feature vector, and consequently the computation time. Feature selection methods employed in this work ensure the effectiveness of the selected feature to perform texture edge detection.

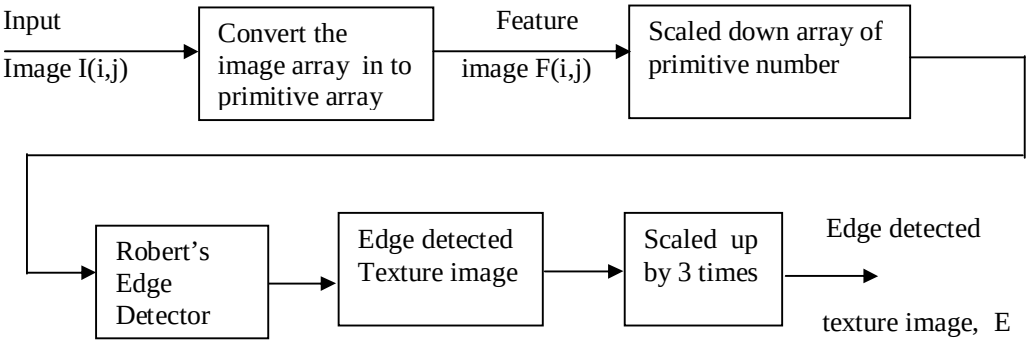


Figure 5.1 Proposed Texture Edge Detection Method for Deterministic Texture Images

In the proposed method, a texture image consists of different texture primitives that are placed regularly or randomly. The images are

characterized by the primitive spectrum where all the texture primitives and their corresponding frequency of occurrences are represented. The texture images may be either deterministic or non-deterministic. In the case of deterministic textures, texture images are considered to consist of different regions, each filled uniformly with a particular primitive. The texture primitives may be present randomly in the case of non-deterministic texture images. The edge detection between different textured regions is performed by adopting suitable techniques as proposed below.

5.3 EDGE DETECTION IN DETERMINISTIC TEXTURED IMAGES

Deterministic textured images have few texture primitives filled uniformly within each region; i.e., each textured region within the texture image has been filled with a particular texture primitive. The edge detection in this image aims to detect the boundaries present between the textured regions. If the conventional edge detection operators are applied like Robert's or Sobel edge operators to the textured images, the micro edges within each primitive will be identified. As a result, a large number of edges will be identified and the discrimination between intra-primitive and within different regions will be a great problem. Because of the difficulty involved in the identification of boundaries between different textured regions by using the conventional edge detection methods, a separate method is presented here for the detection of edges in textured images.

In the previous chapter, it was shown what a set of texture primitives is, and how it is used as a local descriptor. The frequency of the occurrence of the primitives represents the global descriptor. The textured regions are filled only with a particular primitive. By performing a simple conversion of the primitives into the corresponding primitive number, the

image becomes a primitive number array. However, the dimensionality of the array size is reduced three fold when it is compared with the original image size. Because each 3 x 3 image region size is replaced with the corresponding number, only the single digit is used in place of a primitive. This primitive array looks like an array of numbers and any one of the conventional edge detection operators can be applied to detect the boundaries between the textured regions. Now, the Robert's operator (Roberts 1982) is applied and the edges are detected. The detected edges are also smaller in size when compared with the original image size. To overcome this problem, the detected edges are scaled up three fold, and the edges corresponding to the original image is obtained. The entire procedure for detecting the edges in deterministic texture images are given in the form of an algorithm below.

5.3.1 Algorithm for Edge Detection in Deterministic Textured Images

Input : A deterministic texture image I of the size 128 x128

Output : Edge detected image E

Step 1 : Obtain the primitive spectrum for the image I

Step 2 : Identify how many distinct primitives are present and how many regions are there from the primitive spectrum. Both will be the same.

Step 3 : Replace each primitive by the corresponding primitive number for the entire image and generate an array of numbers (array size gets reduced)

Step 4 : Apply the Robert's edge detection operator, namely,

$E_h = | p_1 - p_3| + | p_2 - p_4|$ and

$Ev = |p_1 - p_2 | + | p_3 - p_4 |$ where p_1, p_2, p_3 and p_4 are the

values of the primitive number array and E_h, E_v represent both the horizontal and vertical edges respectively. Overall edge

$Ee = \sqrt{(E_h^2 + E_v^2)}$

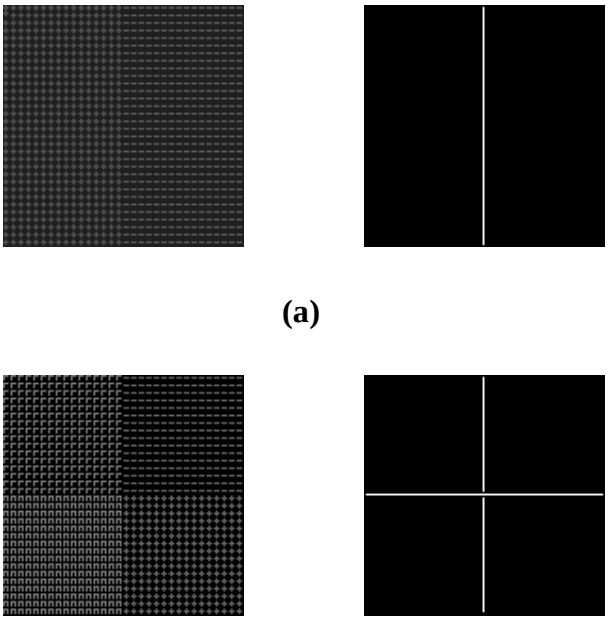
Step 5 : Apply suitable threshold on Ee; the edges are identified

Step 6 : The edge positions are scaled up to coincide with the original image size.

Step 7 : If all the regions are completed, then end with the edge image E

5.3.2 Experimentation and Results

The proposed approach has been tested on many composite texture images, created using the primitives selected from the set of primitives. A few sample deterministic texture images are shown in Figure 5.2. Two images are shown, comprising two or more regions respectively. Edges are detected as per the algorithm described in section 5.3.1. Figure 5.2(a) has two regions, each filled correspondingly by the primitive numbers TPi and TPj. In the first step, these primitives are identified and replaced by the corresponding primitive numbers. The edge detected image is also shown in Figure 5.2 correspondingly. The edges detected are sharp, and coincide accurately with the original image. Similarly, in the other image is shown in Figure 5.2(b), the edge detection results are presented.



(a)

(b)

Figure 5.2 Texture Edge Detection (a to b) for Deterministic Texture Images

5.4 EDGE DETECTION IN NON-DETERMINISTIC TEXTURED IMAGES

The edge detection scheme as suggested in the case of deterministic images can not be directly applied to non-deterministic texture images. The reasons are (i) there is no regularity in the placement of primitives and (ii) various primitives may be present in a single region but the distribution may vary between different textured regions. Hence a separate proposal has been suggested.

In order to have a description of the texture based on local properties, we have used the texture model that has been described in

chapter 3. Each small unit has been represented by the primitive and labeled as the primitive number. The frequency of the occurrence of the primitives is represented by the texture primitive spectrum.

The main criterion for edge detection in textured regions is that the difference in adjacent primitive spectrums of the same textured region is less compared to that of two different textured regions. Finally, the edge is detected by computing the root mean square (rms) value of the following.

$$r = \text{sqrt}\ (r_1^2 + r_2^2)$$

where $$r_1 = \sum_{i=1}^{92} | v_{x,y}(i) - v_{x,y+1}(i) + v_{x+1,y}(i) - v_{x+1,y+1}(i) | \text{ and}$$

$$r_2 = \sum_{i=1}^{92} | v_{x,y}(i) - v_{x+1,y}(i) + v_{x,y+1}(i) - v_{x+1,y+1}(i) |$$

and $v_{x,y}$ is the primitive spectrum of the texture region of the size 30 x 30, centered at x, y coordinates, i varies from 1 to 92. An approximate threshold T is applied to r to detect the presence of an edge. Since image windows of a larger width are considered for computation of the global descriptor called the primitive spectrum for the texture, the detected edges are thickened. The pattern matching thinning algorithm proposed by Chain et al (1987) is applied for thinning the detected textured edges. The algorithms for computation of the local and global descriptors of the texture have been provided in chapter 3. The algorithm for texture edge detection is presented below.

5.4.1 Algorithm for Edge Detection in Non-Deterministic Textured Images

Input : A non-deterministic texture image I of the size 128 x 128

Output : Edge detected image E

Step 1 : Obtain the primitive spectrum for the image I, centered at (x,y),Scanned from top left corner of the texture image, V(x, y)

Step 2 : Repeat step1 for obtaining the spectrums at centers located at (x+1,y), (x, y+1), and (x+1, y+1) for the window size of 30 x 30

Step 3 : Perform edge detection as per the following formula

$$r = \text{sqrt}\left(r_1^2 + r_2^2 \right)$$

where $$r_1 = \sum_{i=1}^{92} | v_{x,y}(i) - v_{x,y+1}(i) + v_{x+1,y}(i) - v_{x+1,y+1}(i) | \text{ and}$$

$$r_2 = \sum_{i=1}^{92} | v_{x,y}(i) - v_{x+1,y}(i) + v_{x,y+1}(i) - v_{x+1,y+1}(i) |$$

Step 4 : Apply threshold T to r on the previous step as,

If r >= T, replace E(x,y) = 255

Else E(x,y) = 0

Step 5 : Edges are identified as E

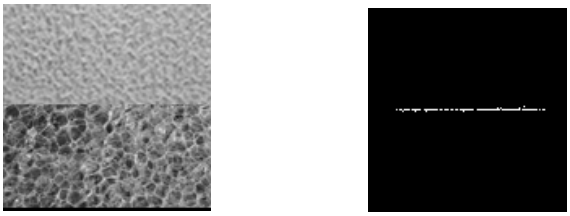
Step 6 : The isolated edge pixels are deleted by comparing with their eight neighbors

Step 7 : The thickened edges are thinned by suitable thinning algorithms on E.

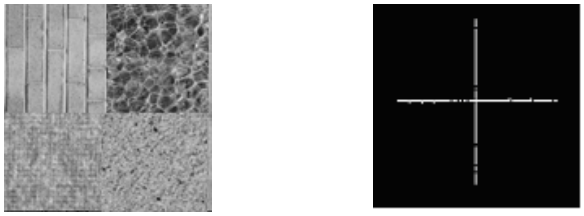
Step 8 : If all the edge pixels are identified, then stop the process.

5.4.2 Experimentation and Results

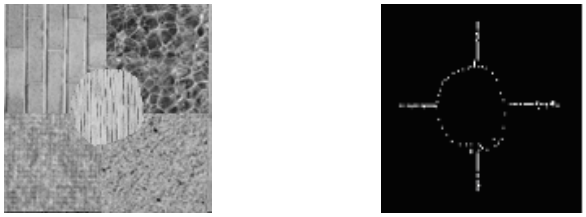
In the case of non-deterministic texture images the edge detection experimentation is as follows. Four test images are considered and shown in Figure 5.3. These images are formed by collecting texture images from the Brodatz album. These images are non-deterministic because there is no uniformity in the placement of texture primitives. The texture primitives and their occurrences are random and non deterministic. Each image is of the size 128 x 128. Figure 5.3(a) has only two textured regions. The algorithm explained in the earlier section 5.4.1 has been applied and the edges identified are also shown correspondingly. Spurious spikes and isolated edge pixels present, are eliminated by suitable post processing of the edge detected image. Figure 5.3(b) has four regions and the edge detected results are also shown. The same is done for the other two images where five and three regions are present. The result of Figure 5.3(d) is presented to show the efficacy of the proposed method for even the irregular edges. The edge detection has been performed using the He and Li Wang (1990) method for the first two images and is shown in Figure 5.3 e.



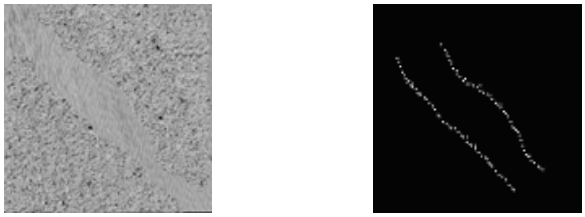
(a)



(b)



(c)



(d)

Figure 5.3 Texture Edge Detection (a to d) for Non-Deterministic Texture Images, (e) using the He and Li Wang's Method

5.5 SUMMARY

The texture edge detection problem has been discussed in this chapter. For texture representation, both local as well as global descriptors have been discussed and used for effective segmentation of textured images using suitable edge detection schemes. The usage of conventional edge detection schemes may not be directly applicable to textured images. Hence the proposed texture primitive spectrum has been shown to be applicable by combining with Roberts's edge detection operator. The results are presented both for the deterministic type and non-deterministic type texture images. Using the edge detected algorithms, the target images have been properly segmented into different regions. The results presented for both the deterministic and non-deterministic texture images reveal the effectiveness of the proposed local as well as global texture descriptors. The application of the global descriptor for effective analysis of skin (textured) images is explained in the next chapter.